

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
20 February 2003 (20.02.2003)

PCT

(10) International Publication Number
WO 03/015288 A1

(51) International Patent Classification⁷: **H03M 13/29**

[US/AU]; 5/9 Quarry Master Road, Pymont, NSW 2009 (AU). XU, Bing [CN/US]; 600 W. Grove Parkway, Apt. 1041, Tempe, AZ 85283 (US).

(21) International Application Number: PCT/US02/24536

(22) International Filing Date: 2 August 2002 (02.08.2002)

(74) Agent: **TEITELBAUM, Ozer, M., N.**; Docket Clerk, Lucent Technologies Inc., P.O. Box 679, Holmdel, NJ 07733-3030 (US).

(25) Filing Language: English

(81) Designated States (*national*): JP, KR, US.

(26) Publication Language: English

(30) Priority Data:
PR6791 3 August 2001 (03.08.2001) AU

(84) Designated States (*regional*): European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).

(71) Applicant (*for all designated States except US*): **LU-CENT TECHNOLOGIES INC.** [US/US]; 600 Mountain Avenue, Murray Hill, NJ 07974-0636 (US).

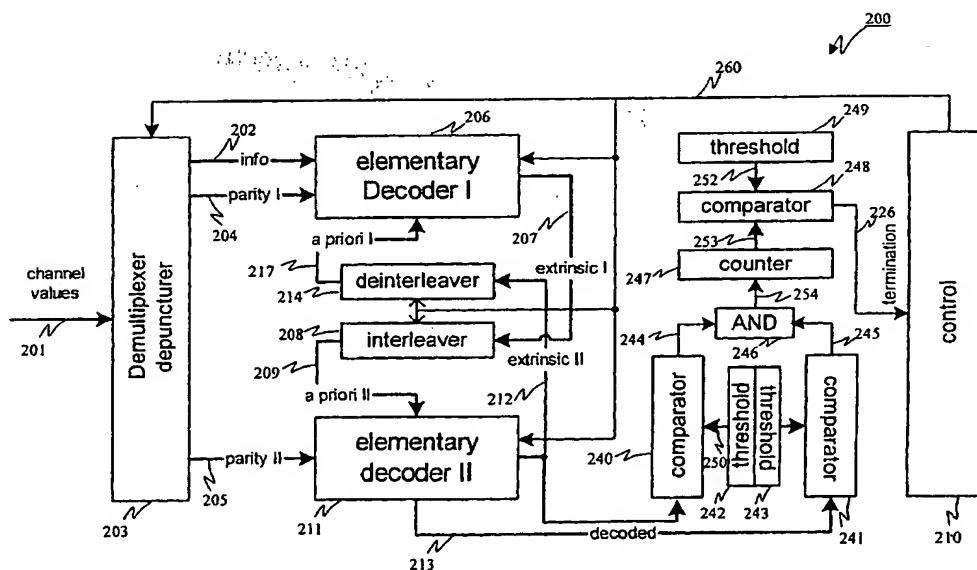
Published:
— with international search report

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **GARRETT, David**

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: **TERMINATION OF ITERATIVE DECODING**



(57) Abstract: A method for terminating iterative decoding. Soft-decision bits and extrinsic information are utilised to determine the convergence of the iterative decoding process. If the extrinsic information is lower than a first predetermined threshold and the soft-decision bits are lower than a second predetermined threshold, a counter is incremented. If the counter is lower than a third predetermined threshold at the end of an iteration, the iterative decoding process is terminated. Otherwise, the counter is reset and a further iteration is commenced.

WO 03/015288 A1

TERMINATION OF ITERATIVE DECODING

CROSS-REFERENCE TO RELATED APPLICATION

5 This application claims priority of Australian Provisional Application No. PR6791, which filed on August 3, 2001.

BACKGROUND OF THE INVENTION

I. FIELD OF THE INVENTION

10 The present invention relates generally to communication systems, and more particularly, to the termination of iterative decoding.

II. DESCRIPTION OF THE RELATED ART

Iterative decoding utilizes a feedback path to present recursive
15 information derived from previous iterations to a current decoding iteration. The current decoding iteration utilizes the recursive information to refine a decoded symbol. The greater the number of iterations employed in the decoding process, the better the bit error rate performance realized. However, there are time and power costs associated with each iteration.
20 Turbo decoding utilizes iterative decoding and random interleaving to achieve an error performance close to the Shannon limit. Consequently, iterative decoding is often employed in channel equalisation and decoding for third generation (3G) mobile communications.

Fig. 1 shows a traditional configuration of a turbo decoder 100.
25 Channel values 101 received by the turbo decoder 100 include systematic data, representing the actual data being transmitted, and parity data, which represents a coded form of the data being transmitted. As seen in Fig. 1, a

demultiplexer 103 receives the channel values 101 and demultiplexes the channel values 101 into systematic data 102, first parity data 101 corresponding to parity data of a first encoder module of a turbo encoder, and second parity data 105 corresponding to parity data of a second encoder module of the same turbo encoder. The demultiplexer 103 presents the first parity data 104 to a first decoder module 106. The first decoder module 106 also receives the systematic data 102 and first *a priori* data 117 from a deinterleaver 114. The first decoder module 106 performs decoding of the systematic data 102 and first parity data using the first *a priori* data 117 to produce first extrinsic data 107.

An interleaver 108 receives the first extrinsic data 107. The interleaver 108 permutes the first extrinsic data 107 with a known bit sequence and produces second *a priori* data 109, which is presented to a second decoder module 111. The first and second *a priori* data 117, 109 passed between the first and second decoder modules 106, 111 provide a measure of probability that bit decisions made by the first and second decoders 106, 111 are correct. In order to generate the required bit probabilities, each of the first and second decoder modules 106, 111 utilizes soft-decision decoding algorithms.

The second decoder module 111 also receives the second parity data 105 from the demultiplexer 103. The second decoder module 111 operates in a manner corresponding to the first decoder module 106 to decode the second *a priori* data 109 and the second parity data 105 to produce second extrinsic data 112 and decoded soft-outputs 113. A deinterleaver 114 receives the second extrinsic data 112 and performs deinterleaving, which is the inverse of the interleaving performed by the interleaver 108, using the same known bit sequence. The deinterleaver 114 produces the first *a priori* data 117, which is presented to the first decoder module 106, as described above.

A control unit 110 presents respective control signals 160, 161, 162, 163, 164 to the demultiplexer 103, first decoder module 106, second decoder module 111, interleaver 108 and deinterleaver 114 so as to afford a recursive

mode of operation. The recursive nature of the turbo decoder 100 ensures that subsequent iterations will improve the probability that the decoded soft-outputs 113 accurately represent an originally transmitted information signal.

The bit error rate performance of turbo codes may, under certain channel conditions, be bounded by the code rate and the interleaver. Once the bit error rate performance approaches this bound, very little further performance improvement may be achieved through performing further iterations. However, further iterations will significantly increase the decoding delay due to the number of additional computations involved. The further iterations will also result in the consumption of additional power resources. Power consumption is a particularly important issue for battery powered mobile terminals. Therefore, certain conditions exist under which once the decoding quality is sufficient for the application at hand, further iterations only give rise to unnecessary computational delay and power consumption.

One approach to reducing the total number of computations, and hence the power consumption, of a turbo decoder focuses on reducing the number of iterations. Once it has been determined that the decoding quality is sufficient for the application at hand, turbo decoding can then be terminated. The early termination of an iterative process may potentially deliver a huge reduction in the amount of computation and, consequently, a reduction in power consumption. The early termination approach is attractive for Very-Large-Scale-Integrated (VLSI) circuit implementation. Two key issues in low-power design for VLSI circuit implementation are termination criteria and termination circuit complexity.

Hagenauer et al in "Iterative decoding of binary block and convolutional codes", *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 429-445, Mar., 1996, proposed that the cross entropy (CE) between the distributions of outputs of the decoder modules 106 and 111 of Fig. 1 is a good measure of decoding quality (decoding convergence). The CE can be used to terminate an iterative decoding process. A turbo encoder typically consists of two constituent

recursive-systematic-convolutional (RSC) encoders. Let u_k and \hat{u}_k be an information bit and an estimate of the information bit, respectively, at time index k . At the i^{th} iteration, let $L_m^{(i)}(\hat{u}_k)$ and $LE_m^{(i)}(\hat{u}_k)$ be the log-likelihood-ratio (LLR) and extrinsic information, respectively, of the estimate of u_k . The index m and i represent a constituent decoder and iteration index, respectively. At iteration i , the CE is defined by the following expression:

$$CE^{(i)} = \sum_{k=1}^N E \left[\log \frac{p_2^{(i)}(\hat{u}_k)}{p_1^{(i)}(\hat{u}_k)} \right] \quad (1)$$

where $p_m^{(i)}(\hat{u}_k)$ is the probability of \hat{u}_k at the output of the m^{th} decoder and N is the information block length. The CE defined above can be used as a measure to stop an iterative decoding process. In order to obtain a simpler expression for CE, the following facts are commonly assumed after the decoding process converges:

- hard decisions (i.e. the sign of the LLR values) at the two constituent decoders are the same and do not change between adjacent iterations;
- the absolute LLR values at the outputs of both constituent decoders are very large, which implies that the decision probabilities are very close to 1;
- the difference of extrinsic information between two adjacent iterations has the same sign as soft decision; and
- the difference between the amplitudes of extrinsic information at adjacent iterations is very small.

Based on the above assumptions, *Hagenauer et al* used an approximated CE to stop the decoding process, which is the well-known *Cross Entropy* termination criterion, and expressed as follows:

$$CE^{(i)} = \sum_{k=1}^N \frac{|\Delta LE_2^{(i)}(\hat{u}_k)|^2}{e^{L_1^{(i)}(\hat{u}_k)}} \quad (2)$$

where $\Delta LE_2^{(i)} = LE_2^{(i)}(\hat{u}_k) - LE_2^{(i-1)}(\hat{u}_k) = L_2^{(i)}(\hat{u}_k) - L_1^{(i)}(\hat{u}_k)$. Although the CE criterion works well to terminate the decoding process with very little BER performance loss, the computation of the CE in equation (2) above is, however, generally considered to be too complicated to be used in VLSI implementation using present technologies.

Known methods for terminating an iterative process using the same CE concept include Sign Change Ratio (SCR) and Hard Decision Aided (HDA) criteria. SCR uses a ratio of sign changes in extrinsic information and information block size as a termination criterion. When the number of sign changes is below a predetermined threshold, the decoding process may be terminated. Such a threshold may, for example, be zero or one bit per block. HDA processing stores hard decision bits from a current decoding iteration. The stored hard decision bits are compared against corresponding hard decision bits of a next iteration. When all hard decision bits of a current iteration agree with the stored hard decision bits of the immediately preceding iteration, the decoding process is terminated. However, SCR and HDA each require a memory equal in size to the block being decoded.

A loglikelihood ratio is the logarithm of the quotient of the probability of receiving a 1 divided by the probability of receiving a 0. A large positive ratio indicates that it is very likely that the bit under consideration is a 1. Conversely, a large negative loglikelihood ratio indicates a high degree of probability that the bit under consideration is a 0. If the loglikelihood ratio is close to 0, the probability that the bit is a 1 is very similar to the probability

that the bit is a 0 and, hence, there is a low degree of confidence in the value of the decoded bit.

In the HDA scheme, a hard decision bit, being the sign of the loglikelihood ratio value, is stored at iteration $m-1$ for each bit to be decoded.

5 Thus, for a code word having n bits, n corresponding memory units are required to store the hard decision bits for a given iteration.

Early termination schemes discussed in previous sections, including HDA and SCR, are devised on the cross-entropy (CE) concept defined in equation (1). Based on the four basic assumptions stated above, the CE is
10 approximated by some cross information between adjacent iterations. For example, CE in equation (2) is defined by the LLR at the first constituent decoder and the difference in extrinsic information between iteration i and $i-1$. In SCR, the cross information on sign changes of extrinsic information between iterations is used as the stopping criterion. In HDA, the cross
15 information on hard decisions between iterations is used as the stopping criterion.

The use of cross information requires memory, because information in a current iteration needs to be stored in order to be compared with information in a successive iteration. Furthermore, termination occurs when
20 there is no change between the last two iterations, so in effect the last iteration is superfluous. Thus, a need exists for a method of terminating an iterative process without requiring comparisons to be made across successive iterations.

25 SUMMARY OF THE INVENTION

In accordance with the principles of the present invention, soft-decision bits and extrinsic information are utilized to determine whether an iterative process has converged and may thus be terminated, without requiring comparisons to be made across successive iterations.

In accordance with one embodiment, extrinsic information and soft-decision bits of a decoder are monitored during an iterative decoding process. At the end of each iteration, the extrinsic information and soft-decision bits are compared against respective first and second predetermined thresholds. Depending on whether the extrinsic information and soft-decision bits are both less than the respective first and second thresholds, a counter is incremented. If the counter is less than a third predetermined threshold, the iterative decoding process is terminated, based on the assumption that the magnitudes of extrinsic information and soft-decision bits are very large after a decoding process has converged. If the counter is greater than the third threshold, it is assumed that the decoding process has not converged and hence may not yet be terminated. Thus, it is no longer necessary to perform a further iteration to determine whether the iterative decoding process has already converged.

In accordance with another embodiment, extrinsic information and soft-decision bits of first and second decoder modules of a decoder are monitored during an iterative decoding process. As extrinsic information and soft-decision bits of both decoder modules are being monitored, it is possible to compare extrinsic information and soft-decision bits against respective first and second predetermined thresholds at the end of an iteration and half-way through an iteration to determine whether the iterative decoding process has converged. The ability to terminate an iterative decoding process half-way through an iteration yields further computational savings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be better understood from reading the following description of non-limiting embodiments, with reference to the attached drawings, wherein below:

FIG. 1 is a block diagram representation of a known turbo decoder;

FIG. 2 is a block diagram representation of one turbo decoder implementing soft-decision and extrinsic aided termination criterion;

FIG. 3 is a block diagram representation of another turbo decoder implementing soft-decision and extrinsic aided termination criterion;

5 FIG. 4 is a flowchart of a process for terminating an iterative process; and

FIG. 5 is a block diagram of a general purpose computer upon which arrangements described can be practised.

It should be emphasized that the drawings of the instant application
10 are not to scale but are merely schematic representations, and thus are not intended to portray the specific dimensions of the invention, which may be determined by skilled artisans through examination of the disclosure herein.

DETAILED DESCRIPTION

15 Termination of a decoding process should advantageously occur as soon as a bit error rate performance desired for the application has been realized. Early termination will eliminate unnecessary computation and reduce power consumption.

Soft-decision and extrinsic aided (SDEA) termination criterion, as it is
20 termed herein, is based on the idea that soft information for a particular decoder has been generated through multiple iterations, meaning that the soft information inherently carries cross information from previous iterations. Thus, the termination can be determined utilizing only information from the current iteration.

25 The original CE in equation (1) defines cross entropy between the distributions of outputs at decoder module one 106 and decoder module two 111 of a turbo decoder 100, which indicates that the cross information between decoder module one 106 and decoder module two 111 should also be applicable to termination criteria. This requires soft-outputs from both
30 constituent decoders. Due to the interleaving operation between the two

constituent decoder modules, the cross information of soft-outputs between decoder module one and decoder module two cannot be obtained without introducing memory. Nevertheless, extrinsic information is passed between the constituent decoder modules and, therefore, the cross information
5 between extrinsic information and soft-outputs at a particular decoder module reflects the cross information between the two constituent decoder modules.

In addition to the assumptions stated above for a converged turbo decoding process, two new assumptions can be added after a decoder
10 converges:

- the magnitudes of extrinsic information and LLR values (soft-outputs) are very large; and
- 15 • at the output of a constituent decoder module, an error bit often happens when the bit's corresponding LLR and soft-output have small magnitudes.

Therefore, Soft Decision and Extrinsic Aided (SDEA) termination criterion
20 provides that at iteration i , the turbo decoding process can be terminated if magnitudes of all loglikelihood ratios and extrinsic information are above first and second predefined thresholds, respectively. Each of the first and second predefined thresholds may be readily determined for a given input precision (the number of bits on the channel data) and internal precision. For example,
25 the thresholds may be readily determined by trial and error since the threshold values are relatively insensitive to noise and channel conditions.

Fig. 2 shows a first arrangement of a turbo decoder 200 implementing soft-decision and extrinsic aided (SDEA) termination criterion. Channel values 201 received by the turbo decoder 200 include systematic data,

representing the actual data being transmitted, and parity data, which represents a coded form of the data being transmitted. A demultiplexer 203 receives the channel values 201 and demultiplexes the channel values 201 into systematic data 202, first parity data 201 corresponding to parity data of a first encoder of a turbo encoder, and second parity data 205 corresponding to parity data of a second encoder of the turbo encoder. The demultiplexer 203 presents the first parity data 204 to a first decoder module 206. The first decoder module 206 also receives the systematic data 202 and first *a priori* data 217 from a deinterleaver 214. The first decoder module 206 performs
10 decoding of the systematic data 202 and first parity data using the first *a priori* data 217 to produce first extrinsic data 207.

An interleaver 208 receives the first extrinsic data 207. The interleaver 208 permutes the first extrinsic data 207 with a known sequence and produces second *a priori* data 209, which is presented to a second decoder module 211.
15 The first and second *a priori* data 217, 209 passed between the first and second decoder modules 206, 211 provides a measure of probability that bit decisions made by the first and second decoder modules are correct. In order to generate the required bit probabilities, each of the first and second decoder modules 206, 211 utilizes soft-decision decoding algorithms.

20 The second decoder module 211 also receives the second parity data 205 from the demultiplexer 203. The second decoder module 211 operates in a manner corresponding to the first decoder module 206 to decode the second *a priori* data 209 and the second parity data 205 to produce second extrinsic data 212 and decoded soft-outputs 213. A first deinterleaver 214 receives the
25 second extrinsic data 212 and performs deinterleaving to produce the first *a priori* data 217, which is presented to the first decoder module 206, as described above. A control unit 210 presents a control signal 260 to each of the demultiplexer 203, first decoder module 206, second decoder module 211, interleaver 208 and deinterleaver 214.

The SDEA criterion utilizes LLR and extrinsic information of the same decoder module at a particular iteration and hence no memory is required. Two comparators are required to check the magnitudes of the LLR and extrinsic information. Accordingly, the second extrinsic data 212 produced by the second decoder module 211 is also presented to a first comparator 240. The first comparator 240 also receives a first threshold 250 from a first threshold unit 242 and determines whether the second extrinsic data 212 is greater than or less than the first threshold 250 to produce an output 244. Similarly, decoded soft-outputs 213 from the second decoder module 211, representing the LLR values, are presented to a second comparator 241, which receives a second threshold 251 from a second threshold unit 243. The second comparator 241 compares the soft decoded outputs 213 against the second threshold 251 to produce an output 245.

The respective outputs 244, 245 from the first and second comparators 240, 241 indicate whether the soft-outputs 213 and the second extrinsic data 212 for each bit being decoded are lower than respective predetermined first and second thresholds 250, 251. The outputs 244, 245 are presented to an AND gate 246, which produces an output 254 to a counter 247. If each of the soft-outputs 213 and the second extrinsic data 212 for the bit being decoded are lower than the respective first and second thresholds 250, 251, the output 254 will be high and consequently the counter 247 will be incremented. Thus, the counter 247 records the number of bits being decoded that have LLR values and extrinsic data lower than the predetermined thresholds.

The counter 247 produces a count 253 to a third comparator 248, which also receives a third threshold 252 from a third threshold unit 249. The third threshold 249 is typically set to zero. However, the third threshold 249 may also be adjusted to satisfy various Bit Error Rate (BER) performance and power requirements. The third comparator 248 compares the count 253 from the counter 247 against the third threshold 252 from the third threshold unit 249 and, if the third threshold 252 is higher than the count 253, a termination

signal 226 is set high and presented to a control unit 210 to terminate the iterative decoding process.

Fig. 3 shows a second arrangement of a turbo decoder 300 implementing soft-decision and extrinsic aided (SDEA) termination criterion. A demultiplexer 303 receives channel values 301 and demultiplexes the channel values 301 into systematic data 302, first parity data 301 corresponding to parity data of a first encoder of a turbo encoder, and second parity data 305 corresponding to parity data of a second encoder of the turbo encoder. The demultiplexer 303 presents the first parity data 304 to a first decoder module 306. The first decoder module 306 also receives the systematic data 302 and first *a priori* data 317 from a deinterleaver 314. The first decoder module 306 performs decoding of the systematic data 302 and first parity data 304 using the first *a priori* data 317 to produce first extrinsic data 307 and first soft decoded outputs 318.

An interleaver 308 receives the first extrinsic data 307. The interleaver 308 permutes the first extrinsic data 307 with a known sequence and produces second *a priori* data 309, which is presented to a second decoder module 311. The first and second *a priori* data 317, 309 passed between the first and second decoder modules 306, 311 provides a measure of probability that bit decisions made by the first and second decoder modules are correct. In order to generate the required bit probabilities, each of the first and second decoder modules 306, 311 utilizes soft-decision decoding algorithms.

The second decoder module 311 also receives the second parity data 305 from the demultiplexer 303. The second decoder module 311 operates in a manner corresponding to the first decoder module 306 to decode the second *a priori* data 309 and the second parity data 305 to produce second extrinsic data 312 and second decoded soft-outputs 313. A first deinterleaver 314 receives the second extrinsic data 312 and performs deinterleaving to produce the first *a priori* data 317, which is presented to the first decoder module 306, as described above.

A first multiplexer 315 receives each of the first extrinsic data 307 and the second extrinsic data 312 and produces selected extrinsic data 319, depending on which of the first and second decoder modules 306, 311 is active. Similarly, a second multiplexer 316 receives each of the first and second soft decoded outputs 318 and 319, respectively, to produce selected soft-outputs 320. The selected extrinsic data 319 is presented to a first comparator 340. The first comparator 340 also receives a first threshold 350 from a first threshold unit 342, and determines whether the selected extrinsic data 319 is greater than or less than the first threshold 350 to produce an output 344. Similarly, selected soft-outputs 320, representing the LLR values of whichever of the first and second decoder modules 306, 311 is active, are presented to a second comparator 341, which receives a second threshold 351 from a second threshold unit 343. The second comparator 341 compares the selected soft-outputs 320 against the second threshold 351 to produce an output 345.

The respective outputs 344, 345 from the first and second comparators 340, 341 indicate whether the selected soft-outputs 320 and the selected extrinsic data 319 for each bit being decoded are higher than respective predetermined first and second thresholds 350, 351. The outputs 344, 345 are presented to an AND gate 346, which produces an output 354 to a counter 347. If each of the selected soft-outputs 320 and the selected extrinsic data 319 for the bit being decoded are lower than the respective first and second thresholds 350, 351, the output 354 will be high and consequently the counter 347 will be incremented. Thus, the counter 347 records the number of bits being decoded that have LLR values and extrinsic data lower than the predetermined thresholds.

The counter 347 produces a count 353 to a third comparator 348, which also receives a third threshold 352 from a third threshold unit 349. The third threshold 349 is typically set to zero. However, the third threshold 349 may also be adjusted to satisfy various Bit Error Rate (BER) performance and

power requirements. The third comparator 348 compares the count 353 from the counter 347 against the third threshold 352 from the third threshold unit 349 and, if the third threshold 352 is higher than the count 353, a termination signal 326 is set high and presented to a control unit 310 to terminate the iterative decoding process. As decoded soft outputs 318, 313 and extrinsic data 307, 312 are monitored from the respective first and second decoder modules 306, 311, it is possible to terminate the decoding process half-way through a decoding iteration if the quality of the decoding is determined to be sufficient for the application at hand. Thus, the arrangement of Fig. 3 may be used to reduce the number of computations by half an iteration and thus produce a greater saving in power consumption.

Typically, the first decoder module 306 and the second decoder module 311 operate in alternate time periods to process a stream of input data. When the first decoder module 306 is decoding the systematic data 302 and first parity data 304 using the first *a priori* data 317 to produce first extrinsic data 307 and first soft decoded outputs 318, the second decoder module 311 is idle. Conversely, when the second decoder module 311 is decoding the second *a priori* data 309 and the second parity data 305 to produce second extrinsic data 312 and second decoded soft-outputs 313, the first decoder module 306 is idle. Thus, it is possible to implement the first and second decoder modules 306, 311 using a single decoder module that changes functionality in alternating time periods to process the relevant inputs for that time period.

Similarly, the interleaver 308 and the deinterleaver 314 typically operate in alternate time periods. Therefore, it is possible to implement the interleaver 308 and the deinterleaver 314 using a single interleaving module that changes functionality in alternating time periods to manipulate incoming decoded soft-outputs appropriately.

The SDEA method is less complex than other termination schemes for iterative decoding. In the case where a very low BER performance is required

(<10⁻⁶), a first decoding iteration is used to test the initial threshold and the SDEA process commences at the second iteration, based on a refined threshold. The cost of the additional decoding iteration is traded for lower BER performance.

5 The SDEA criterion has the further advantage that while providing good performance, on par with HDA and 6 iterations of turbo decoding, it results in a reduction in the average number of iterations. The reduction in average iterations for SDEA show that SDEA has a measure of termination that is a much better predictor of convergence of the turbo decoding process.

10 Tests performed indicate that SDEA arrangements described herein perform better than prior hard decision based methods for termination in both BER and power performance without the memory requirement for comparisons between iteration. In particular, the SDEA BER performance is very close to that obtained by 6 iterations of turbo decoding, but with fewer
15 iterations required. SDEA reduces the average number of turbo iterations from 6 iterations down to 2 iterations at a signal-to-noise ratio (SNR) of 3 dB, a 66% power savings over regular turbo decoding with 6 iterations. In the SNR range between 0 dB and 3 dB, SDEA can save over half an iteration in most cases over the well-known HDA early termination algorithm.

20 Fig. 4 shows a flowchart 400 of a process for terminating an iterative decoding process. The steps of Fig. 4 are practised on blocks of data. The iterative decoding process commences at step 410 and proceeds to step 420. Step 420 demultiplexes received channel values to produce systematic data, first parity data and second parity data. Step 430 decodes the systematic data
25 and the first parity data, using first *a priori* data, to produce first extrinsic data. Step 440 interleaves the first extrinsic data to produce second *a priori* data. Step 450 decodes the second parity data and the second *a priori* data to produce second extrinsic data and soft-outputs.

30 The process continues to step 460, which deinterleaves the second extrinsic data to produce the first *a priori* data. Step 470 determines whether

or not each instance of data in the second extrinsic data is lower than a first predetermined threshold and whether a corresponding soft-output is lower than a second predetermined threshold. If both of these conditions are met, a counter is incremented in step 475 and the process continues to step 480. If
5 either one or both of the conditions are not satisfied in step 470, step 475 is bypassed and the process proceeds to step 480.

Step 480 determines whether each instance of data in the block of second extrinsic data has been compared against the first predetermined threshold. If the entire block of data has been processed, the process
10 continues to step 485. If there are instances of data in the block of second extrinsic data that have not yet been compared, the process returns to step 470. Step 485 determines whether the counter is less than a third threshold. If the counter is less than the third threshold, the iterative decoding process terminates at step 495. If, however, the counter is greater than the third
15 threshold, the counter is reset in step 490 and the process repeats from step 420.

The method of Fig. 4 may be practised using a general-purpose computer system 500, such as that shown in Fig. 5 wherein the processes of Fig. 4 are implemented as software, such as an application program executing
20 within the computer system 500. In particular, the steps of method of Fig. 4 are effected by instructions in the software that are carried out by the computer. The instructions may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part performs the Fig. 4
25 methods and a second part manages a user interface between the first part and the user. The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer from the computer readable medium, and then executed by the computer. A computer readable medium having such
30 software or computer program recorded on it is a computer program product.

The use of the computer program product in the computer advantageously effects an advantageous apparatus for Fig. 4.

The computer system 500 comprises a computer module 501, input devices such as a keyboard 502 and mouse 503, output devices including a printer 515 and a display device 514. A Modulator-Demodulator (Modem) transceiver device 516 is used by the computer module 501 for communicating to and from a communications network 520, for example connectable via a telephone line 521 or other functional medium. The modem 516 can be used to obtain access to the Internet, and other network systems, such as a Local Area Network (LAN) or a Wide Area Network (WAN).

The computer module 501 typically includes at least one processor unit 505, a memory unit 506, for example formed from semiconductor random access memory (RAM) and read only memory (ROM), input/output (I/O) interfaces including a video interface 507, and an I/O interface 513 for the keyboard 502 and mouse 503 and optionally a joystick (not illustrated), and an interface 508 for the modem 516. A storage device 509 is provided and typically includes a hard disk drive 510 and a floppy disk drive 511. A magnetic tape drive (not illustrated) may also be used. A CD-ROM drive 512 is typically provided as a non-volatile source of data. The components 505 to 513 of the computer module 501, typically communicate via an interconnected bus 504 and in a manner, which results in a conventional mode of operation of the computer system 500 known to those in the relevant art. Examples of computers on which the described arrangements may be practised include IBM-PC's and compatibles, Sun Sparcstations or computer systems evolved therefrom.

Typically, the application program is resident on the hard disk drive 510 and read and controlled in its execution by the processor 505. Intermediate storage of the program and any data fetched from the network 520 may be accomplished using the semiconductor memory 506,

possibly in concert with the hard disk drive 510. In some instances, the application program may be supplied to the user encoded on a CD-ROM or floppy disk and read via the corresponding drive 512 or 511, or alternatively may be read by the user from the network 520 via the modem device 516. Still
5 further, the software can also be loaded into the computer system 500 from other computer readable media. The term "computer readable medium" as used herein refers to any storage or transmission medium that participates in providing instructions and/or data to the computer system 500 for execution and/or processing. Examples of storage media include floppy disks,
10 magnetic tape, CD-ROM, a hard disk drive, a ROM or integrated circuit, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 501. Examples of transmission media include radio or infra-red transmission channels as well as a network connection to another
15 computer or networked device, and the Internet or Intranets including email transmissions and information recorded on websites and the like.

The method of Fig. 4 may alternatively be implemented in dedicated hardware such as one or more integrated circuits performing the functions or sub functions of Fig. 4. Such dedicated hardware may include graphic
20 processors, digital signal processors, or one or more microprocessors and associated memories.

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have for the purposes of this description the same
25 function(s) or operation(s), unless the contrary intention appears.

It is apparent from the above that the arrangements described are applicable to iterative decoding in communications systems.

While the particular invention has been described with reference to illustrative embodiments, this description is not meant to be construed in a
30 limiting sense. It is understood that although the present invention has been

described, various modifications of the illustrative embodiments, as well as additional embodiments of the invention, will be apparent to one of ordinary skill in the art upon reference to this description without departing from the spirit of the invention, as recited in the claims appended hereto.

5 Consequently, the method, system and portions thereof and of the described method and system may be implemented in different locations, such as a wireless unit, a base station, a base station controller, a mobile switching center and/or a radar system. Moreover, processing circuitry required to implement and use the described system may be implemented in application
10 specific integrated circuits, software-driven processing circuitry, firmware, programmable logic devices, hardware, discrete components or arrangements of the above components as would be understood by one of ordinary skill in the art with the benefit of this disclosure. Those skilled in the art will readily recognize that these and various other modifications, arrangements and
15 methods can be made to the present invention without strictly following the exemplary applications illustrated and described herein and without departing from the spirit and scope of the present invention. It is therefore contemplated that the appended claims will cover any such modifications or embodiments as fall within the true scope of the invention.

20

WE CLAIM:**1. A method comprising:**

5 determining a convergence from at least one block of extrinsic data and soft outputs to terminate an iterative decoding step.

10 2. The method of Claim 1, wherein the iterative decoding step is terminated half-way through.

3. The method of Claim 1, further comprising:

15 incrementing a counter upon comparing the at least one block of extrinsic data with a first threshold and/or upon comparing the corresponding soft output with a second threshold.

20 4. The method of Claim 3, further comprising:

terminating the iterative decoding step upon comparing the counter with a third threshold and/or resetting the counter upon comparing the counter with the third threshold.

25

5. The method of Claim 4, further comprising:

30 reinitiating the iterative decoding step in response to resetting the counter.

6. The method of Claim 3, wherein said first and second thresholds are dependent upon an input precision and internal precision of the iterative decoding, and the third threshold is set to zero.

5

7. The method of Claim 1, wherein the iterative decoding step comprises at least a first and a second turbo decoding step.

10 8. The method of Claim 1, further comprising:

interleaving an earlier block of data to generate a block of *a priori* data;

15 decoding at least the block of *a priori* data to generate the at least one block of extrinsic data and soft outputs; and

deinterleaving the at least one block of extrinsic data to produce a new block of first *a priori* data for a next step in iterative decoding.

20

9. A method of iterative decoding comprising:

demultiplexing channel values to produce a block of systematic data, first parity data and second parity data;

25

decoding said block of systematic data and said first parity data using a block of first *a priori* data to produce a block of first extrinsic data and first soft-outputs;

incrementing a counter for each instance of data in said block of first extrinsic data if said instance of first extrinsic data is lower than a first predetermined threshold and said corresponding first soft output is lower than a second predetermined threshold;

5

terminating said iterative decoding if said counter is less than a third predetermined threshold or, if said counter is greater than said third predetermined threshold, resetting said counter;

10

interleaving said block of first extrinsic data to produce a block of second *a priori* data;

15

decoding said block of second parity data and said block of second *a priori* data to produce a block of second extrinsic data and second soft-outputs;

deinterleaving said block of second extrinsic data to produce a new block of first *a priori* data for a next iteration;

20

incrementing a counter for each instance of data in said block of second extrinsic data if said instance of second extrinsic data is lower than a fourth predetermined threshold and said corresponding second soft-output is lower than a fifth predetermined threshold; and

25

terminating said iterative decoding if said counter is less than a sixth predetermined threshold, or resetting said counter and repeating steps (a) to (h), if said counter is greater than said sixth predetermined threshold.

10. The method of Claim 9, wherein said predetermined thresholds are selected from the group consisting of:

the first threshold is equal to said fourth threshold;

the second threshold is equal to said fifth threshold; and

the third threshold is equal to said sixth threshold.

11. The method of Claim 9, wherein said third threshold is set to zero.

12. A method of iterative decoding comprising:

for each instance of data in a block of data:

comparing a loglikelihood ratio with a first threshold produced by the iterative decoding step in a current iteration;

comparing extrinsic information with a second threshold produced by the iterative decoding step in a current iteration;

incrementing a counter if the loglikelihood ratio is lower than the first threshold and the extrinsic information is lower than the second threshold; and

terminating the iterative decoding step if the counter is less than or equal to a third threshold; or

resetting the counter and repeating the comparing steps and the incrementing step if the counter is greater than the third threshold.

5 13. The method of Claim 13, wherein the third threshold is set to zero.

14. An iterative apparatus comprising:

10 a demultiplexer for demultiplexing channel values to produce a block of systematic data, first parity data and second parity data;

15 a first decoder for decoding the block of systematic data and the first parity data, using a block of first *a priori* data, to produce a block of first extrinsic data;

an interleaver for interleaving the block of first extrinsic data to produce a block of second *a priori* data;

20 a second decoder for decoding the block of second parity data and the block of second *a priori* data to produce a block of second extrinsic data and soft-outputs;

25 a deinterleaver for deinterleaving the second extrinsic data to produce a new block of said first *a priori* data for a next iteration;

a counter, which is incremented if the instance of second extrinsic data is lower than a first threshold and the corresponding soft-output is lower than a second threshold; and

a comparator for generating a terminating signal if the value of the counter is less than a third threshold, and/or for resetting the counter if the counter is less than the third threshold.

5

15. The iterative apparatus of Claim 14, wherein the third predetermined threshold is set to zero.

10 16. The iterative apparatus of Claim 14, wherein the first and second decoders are implemented using a single decoder module.

17. The iterative apparatus of Claim 16, wherein the single decoder module
15 is reconfigurable for operating as the first decoder and/or second decoder.

18. The iterative apparatus of Claim 14, wherein the interleaver and the deinterleaver are implemented by a single interleaving module.

20

19. The iterative apparatus of Claim 18, wherein the single interleaving module is reconfigurable for operating as the interleaver and/or the deinterleaver.

25

20. The iterative apparatus of Claim 14, wherein the apparatus implements a turbo decoding function.

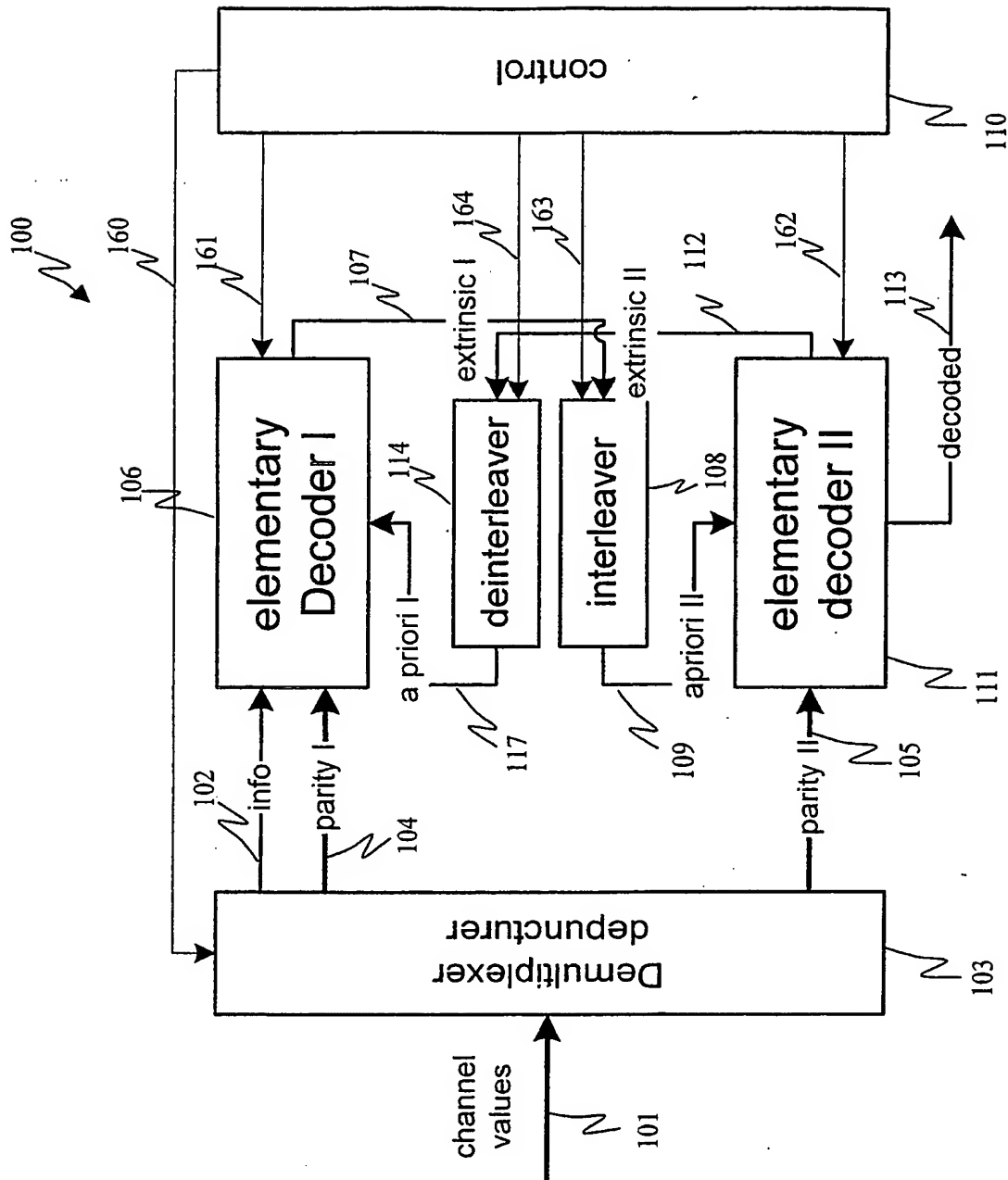


Fig. 1
(PRIOR ART)

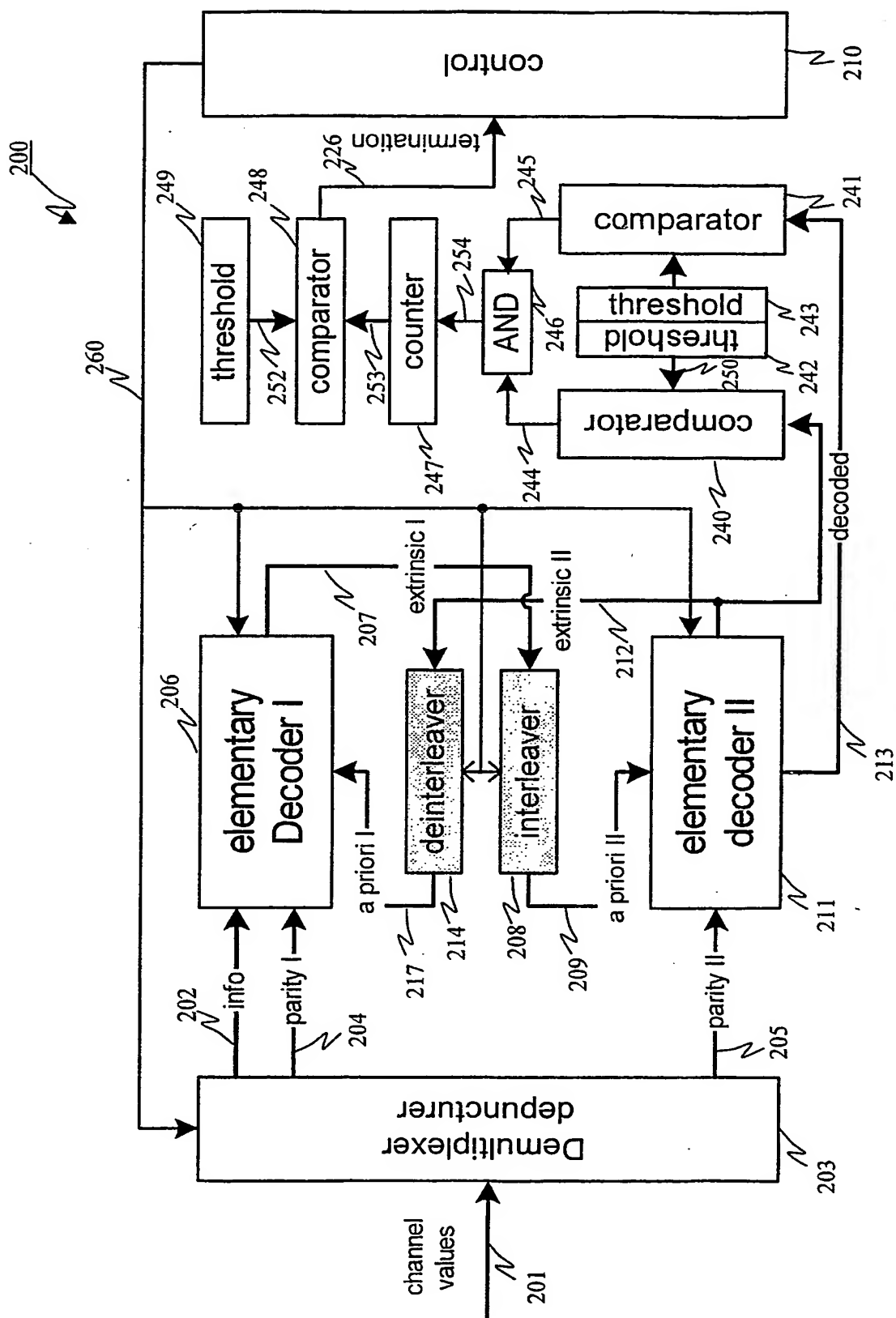


Fig. 2

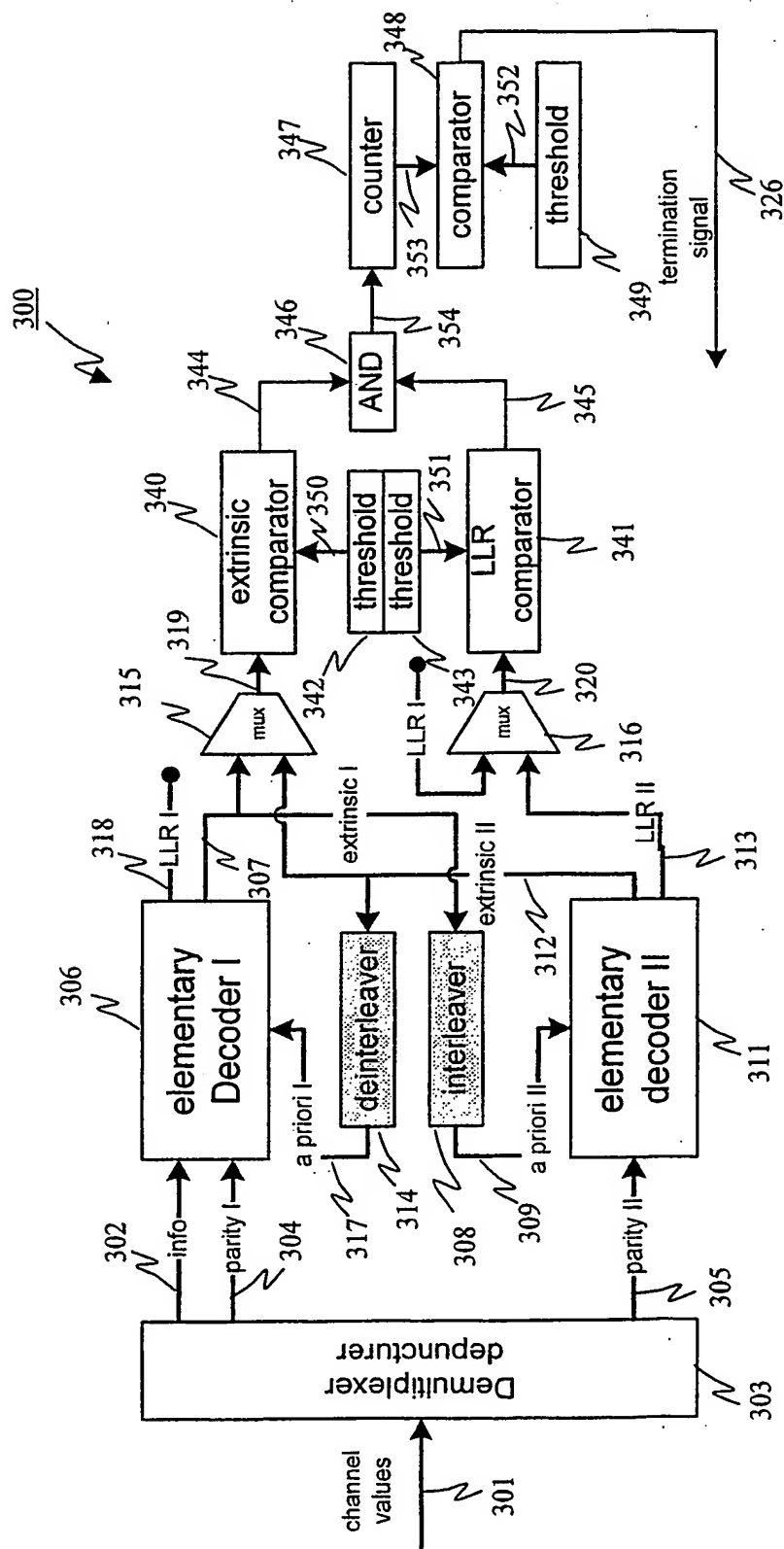


Fig. 3

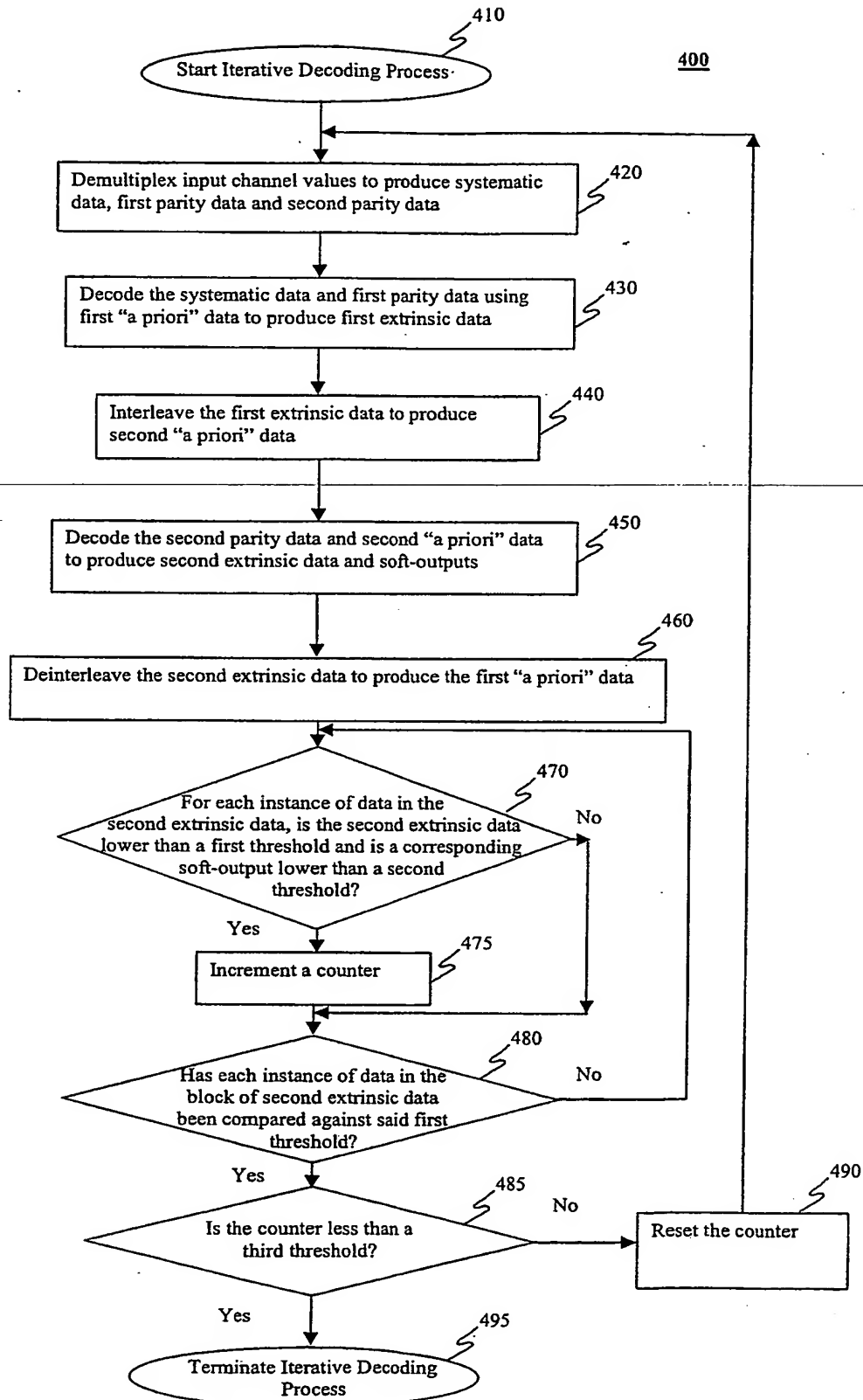
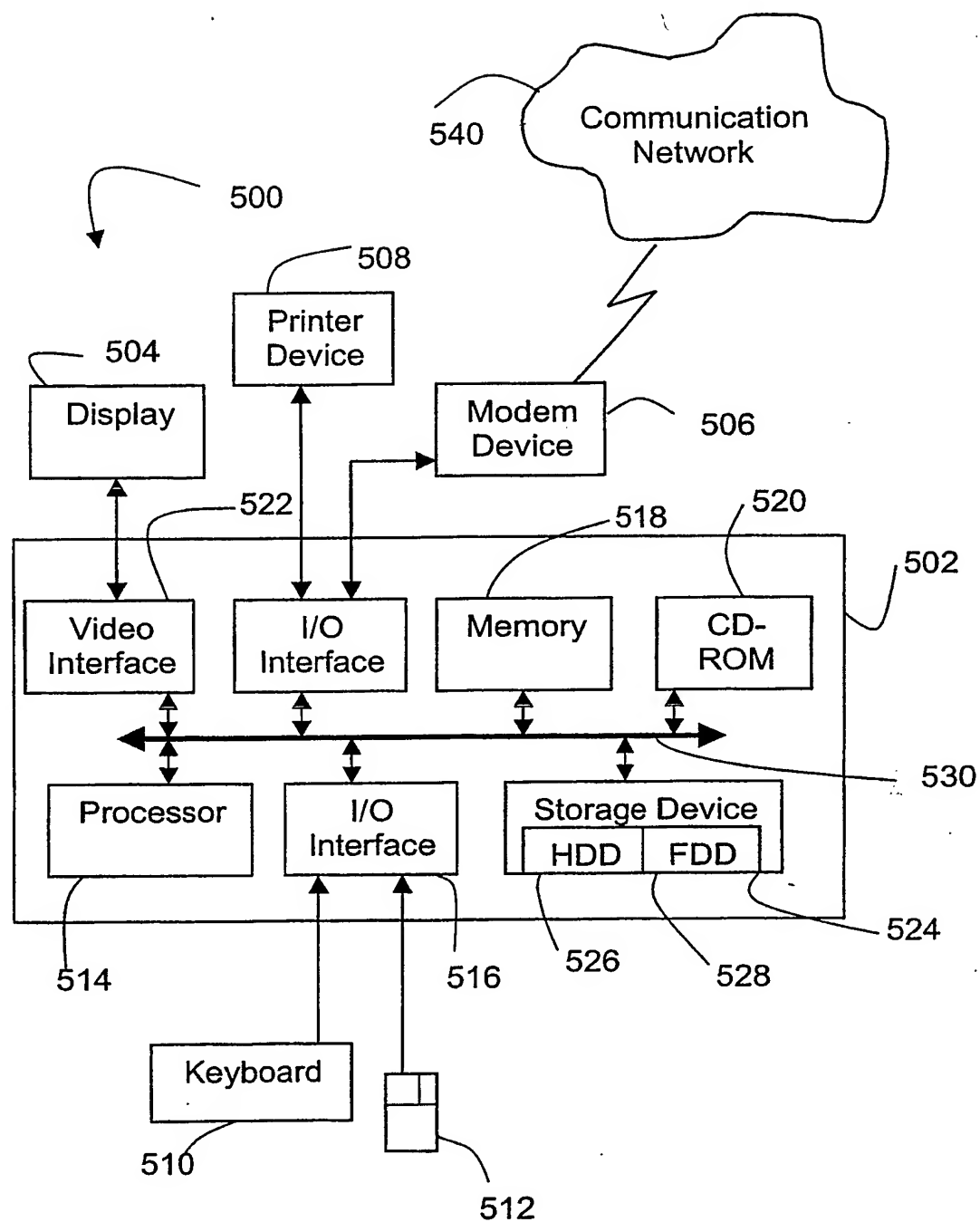
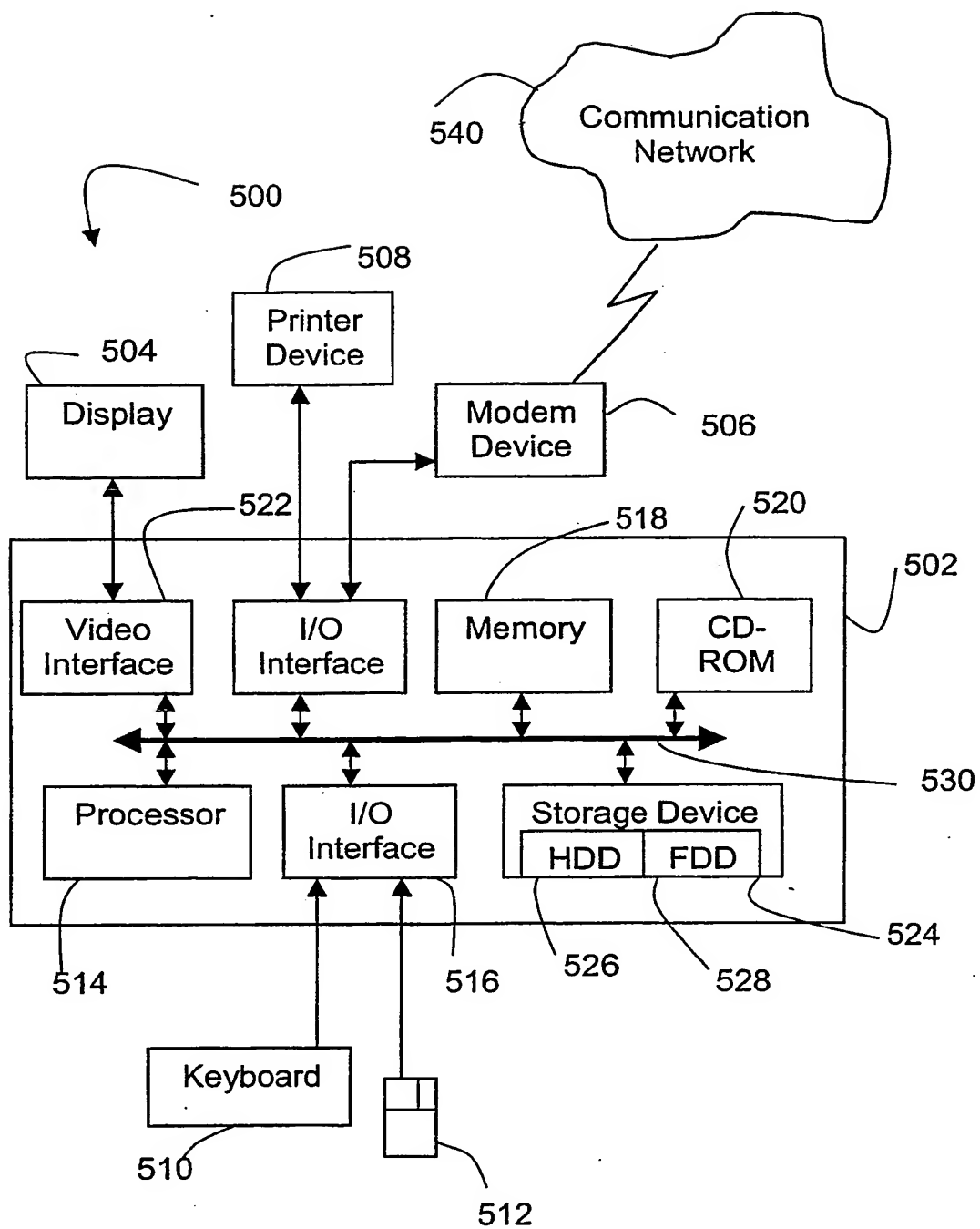


Fig. 4

**Fig. 5**

**Fig. 5**

INTERNATIONAL SEARCH REPORT

PCT/US 02/24536

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X	DAVID GARRETT, BING XU, CHRIS NICOL: "Energy efficient Turbo Decoding for 3G Mobile" ISLLPED '01: PROCEEDING OF THE 2001 INTERNATIONAL SYMPOSIUM ON LOW POWER ELECTRONICS AND DESIGN, 6 - 7 August 2001, pages 328-333, XP002215915 Huntington Beach, CA, USA page 328 -page 333 -----	1-15,20